

(19) The Japanese Patent Office  
(12) Laid-Open Patent Application Publication (A)  
(11) Laid-Open Patent Application Publication No. P2002-341859A  
(43) Publication Date: November 29, 2002 (2002.11.29)  
(51)

Int. Cl. <sup>7</sup>	Classification Symbol	FI
G09G 5/38		G09G 5/38
A63F 13/00		A63F 13/00
		B 2C001
		C 5B080
		D 5C082
G06T 11/00	100	G06T 11/00 100A

Request for Examination: Requested

Number of Claims:4 OL (6 pages in total)

(21) Application No. P2001-148135  
(22) Application Date: May 17, 2001 (2001.5.17)  
(71) Applicant: 000004075

Yamaha Corp.

No. 10-1 Nakazawa-cho, Hamamatsu-shi,  
Shizuoka-ken, Japan

(72) Inventor: Shuhei ITO

c/o Yamaha Corp.,

No. 10-1 Nakazawa-cho, Hamamatsu-shi,  
Shizuoka-ken, Japan

(74) Agent: 100064908

Patent Attorney, Masatake SHIGA

F-Term: 20001	BC00	BC05	BC06			
5B080	GA01	CA08	FA08	FA09		
5O082	AA06	BA34	BA43	BB15	BB32	
	BB44	BB53	DA73	MM02	MM07	

(54) [Title of the Invention] IMAGE DISPLAY DEVICE

(57) [Abstract]

[Problem to be Solved] To provide an image display device that can

improve display capacity without requiring large cost increase and without damaging the reliability.

[Solution] Data for displaying sprites are compressed and stored in a pattern ROM 25. Attribution data for designating display positions or the like within the ROM 25 are stored in a sprite attribution table 23. The sprite data are read out from the ROM 25 in accordance with the attribution data and are decompressed by a real-time decompression processing circuit 30. The decompressed data are converted to RGB data to be written into a sprite buffer 33. The data within the buffer 33 are written into a storing area of a frame buffer 39 designated by the attribution data. The data within the frame buffer 39 are read out in order and in accordance with display scan timing to be displayed at a display device 26.

[Claims]

[Claim 1]

An image display device comprising:

a sprite attribution table into which sprite attribution data are written;

reading out means for reading out sprite pattern data compressed in a piece of memory, in accordance with designations of the sprite attribution table;

decompressing means for decompressing the sprite pattern data read out from the piece of memory to the original state that was before being compressed;

a sprite buffer into which the data decompressed by the decompressing means are written;

writing means for reading out the data stored in the sprite buffer and for writing the data into a storing area of a frame buffer designated by the sprite attribution data; and

display means for displaying the data stored in the frame buffer.

[Claim 2]

The image display device according to Claim 1, wherein the decompressing means further converts the decompressed data into RGB data in accordance with data formats read out from the sprite attribution table, and outputs the RGB data.

[Claim 3]

The image display device according to Claim 1 or 2, wherein the writing means alters the storing area of the frame buffer, which is designated by the sprite attribution data, in accordance with a predetermined rule to write the data outputted from the sprite buffer into the altered area.

[Claim 4]

The image display device according to any one of Claims 1 to 3, wherein the writing means writes into the frame buffer in a manner

of overwriting.

[Detailed Description of the Invention]

[0001]

[Field of the Invention]

The present invention relates to an image display device whereby improvement of sprite display performance is intended.

[0002]

[Background Art]

As is well known, a sprite display system is often used in a field of video games or the like. In this connection, the sprite display system is a system such that a display position attribution (sprite attribution) is allotted every character displayed in a picture and the characters are arranged in accordance with the sprite attributions to configure the entire picture. The character having the display position and so forth is called "sprite." In a case such as, for example, a video game, in which characters are interactively and rapidly moved, the sprite display system allows a picture to be altered solely by modifying the sprite attributions of the respective characters to be moved.

[0003]

FIG. 3 is a block diagram showing a structure of an image display device configured in accordance with a conventional sprite display system. A sprite attribution table 1 stores attribution data regarding sprites, such as, for example, a display position, a sprite pattern (character's design) number, and a data format of each sprite pattern data (design data) each per sprite displayed in a display device 2. Based upon the sprite pattern numbers stored in the sprite attribution table 1, a pattern number reading out circuit 4 and a pattern ROM address generating circuit together generate addresses of a pattern ROM 6 at which the respective pattern data are stored.

[0004]

On the other hand, a display position reading out circuit 8 reads out the sprite display data stored in the sprite attribution table 1 and outputs them to a frame buffer address generating circuit 9. The frame buffer address generating circuit 9 generates frame buffer addresses based upon the sprite display position data, and outputs them to a frame buffer 10. The pattern data read out from the pattern ROM 6 are stored into the frame buffer 10 in accordance with the addresses.

[0005]

Reading processes of the pattern data and writing processes thereof to the frame buffer 10 are executed in corresponding to the number of the displayed sprites so that the data for one picture are laid in the frame buffer. The pattern data stored in the frame buffer 10 are read out from the frame buffer 10 in accordance with scan timing of the display device 2, and are outputted to a display data processing section 12. The display data processing section 12 outputs the pattern data to a color look-up table 13 or a YUV decoder 14 in accordance with the pattern data formats read out from the sprite attribution table; thereby, the display data processing section 12 converts the pattern data into RGB (red/green/blue) data, and outputs them to the display device 2.

[0006]

[Problem to be Solved by the Invention]

Meanwhile, in the image display device described above, the number of sprite displays is limited according to an access speed of the pattern ROM 6. That is, the frame buffer 10 refreshes (alters image data) synchronously with scan periods of the display device 2. Normally, one frame has the period of 60Hz. If the access speed of the mask ROM is 125nS (8MHz) and a bus width is 8bits, a data amount to be transmitted by the one frame is:

$$8\text{MByte} / 60\text{Hz} = 133\text{Kbyte}.$$

If the number of sprite colors is sixteen (data amount is 4bits

per dot),

$133K * 8/4 = 266 * 1024$  dots (approximately 272000 dots).

If resolution 320 x 240, which is general in NTSC, is applied, a sprite display capacity:

$272000 / (320 * 240) =$  approximately 3.5 frames

can be obtained. This, however, is not sufficient display capacity if an animation display in which characters are moved in a background pattern (also depicted with sprites) is supposed, and overlaps of some sprites with each other are taken into consideration.

[0007]

Accordingly, some ideas that can improve the display capacity are necessary. In order to improve the display capacity, the following manners can be employed.

- (a) The bus width of the interface with the pattern ROM 6 is expanded.
- (b) The pattern data of the pattern ROM 6 is downloaded to, for example, SDRAM that can make a rapid transfer so as to realize speedup of access cycles.

[0008]

However, the manner (a) described above can cause a problem that the number of terminals for the interface with the pattern ROM 6 which is a separate chip increases, and thus invites cost increase. Also, the pattern ROM 6 needs to be formed with multiple ROMs connected parallel to each other. This is another reason for the cost increase. Meanwhile, the manner (b) requires that the SDRAM be attached externally. Cost increase due to the SDRAM cost is thus resulted. Also, an interface with the SDRAM is required and the number of terminals for the interface necessarily increases. Further, because the pattern data are transferred to the volatile memory (SDRAM), the reliability of the system can be damaged. That is, the volatile memory can malfunction by disturbance noises such as, for example, static electricity. In addition, the volatile memory can cause problems such as, for example, data errors. It

is one idea that the SDRAM is built in. However, patterns through the entire scenario need capacity of 64Mbits-256Mbits and thus it has no reality. There can be another manner that decreases the capacity of the SDRAM because only patterns to be used every scene are downloaded to be stored. This manner, however, has problems such that download is needed every scene and the number of patterns that can be used at one scene is limited.

[0009]

The present invention is made under the circumstances and aims to provide an image display device that can improve sprite display capacity without requiring large cost increase and without damaging the reliability.

[0010]

[Means for Solving the Problem]

The invention is made to solve the problems discussed above. The invention recited in Claim 1 is an image display device including: a sprite attribution table into which sprite attribution data are written; reading out means for reading out sprite pattern data compressed in a piece of memory, in accordance with designations of the sprite attribution table; decompressing means for decompressing the sprite pattern data read out from the piece of memory to the original state that was before being compressed; a sprite buffer into which the data decompressed by the decompressing means are written; writing means for reading out the data stored in the sprite buffer and for writing the data into a storing area of a frame buffer designated by the sprite attribution data; and display means for displaying the data stored in the frame buffer.

[0011]

The invention recited in Claim 2 is the image display device according to Claim 1, in which the decompressing means further converts the decompressed data into RGB data in accordance with

data formats read out from the sprite attribution table, and outputs the RGB data. The invention recited in Claim 3 is the image display device according to Claim 1 or 2, in which the writing means alters the storing area of the frame buffer, which is designated by the sprite attribution data, in accordance with a predetermined rule to write the data outputted from the sprite buffer into the altered area.

[0012]

The invention recited in Claim 4 is the image display device according to any one of Claims 1 to 3, in which the writing means writes into the frame buffer in a manner of overwriting.

[0013]

[Embodiment of the Invention]

With reference to drawings, one embodiment of the invention will be described below. FIG. 1 is a block diagram showing a structure of an image display device 21 configured in accordance with the embodiment. FIG. 2 is a block diagram showing a structure of a display unit formed with the image display device 21 and a CPU (central processing unit) 22. In FIG. 2, a pattern ROM 25 (mask ROM) stores pattern data of multiple sprites by being compressed using a compression manner such as, for example, a run-length compression manner which allows real-time decompressing. The CPU 22 outputs display position data and other sprite attribution data to the image display device 21. The display position data designate which positions in a displayed picture are allotted to the respective sprites stored in the pattern ROM 25 for showing them. The image display device 21 reads out the pattern data of the respective sprites from the pattern ROM 25 and makes real-time decompression thereof. The image display device 21 further converts them into RGB data and has the display device 26 display them in accordance with position data sent from the CPU 22.

[0014]



The embodiment will be described in detail below. In FIG. 1, reference numeral 23 indicates a sprite attribution table configured with a RAM (random access memory). Attribution data indicative of attribution of the sprites stored in the pattern ROM 25 are written into the sprite attribution table 23 per sprite by the CPU 22. The attribution data include pattern numbers indicative of numbers of the sprite patterns, display position data designating the display positions of the sprites on the displayed pictures, data formats of the sprite pattern data, pallet numbers designating numbers of color pallets to be used, and so forth.

[0015]

Reference numeral 28 indicates a pattern number reading out circuit that reads out the pattern numbers stored in the sprite attribution table in numeric order from one which is subordinate in priority of the display relative to others. Reference numeral 29 indicates a pattern ROM address generating circuit that generates addresses for reading out the sprites from the pattern ROM 25 upon receiving the pattern numbers read out from the pattern number reading out circuit 28. The pattern ROM address generating circuit 29 outputs the generated addresses to the pattern ROM 25. When an address described above is supplied to the pattern ROM 25, a piece of the pattern data of the sprites is read out from the pattern ROM 25 and is outputted to a real-time decompression processing circuit 30.

[0016]

The real-time decompression processing circuit 30 decompresses compressed pattern data read out from the pattern ROM 25 to the original state that was before being compressed, by a real-time processing. The real-time decompression processing circuit 30 then converts the decompressed data into the RGB data using a color look-up table 31 or a YUV decoder 32 to write them into a sprite buffer 33. In this connection, whether the color look-up table 31

is used or the YUV decoder 32 is used is determined depending upon the data formats read out from the sprite attribution table 23. That is, if the format of the pattern data before the compression has been the color code, the color look-up table 31 is used. Meanwhile, if the format has been the YUV data, the YUV decoder 32 is used.

[0017]

On the other hand, a display position reading out circuit 35 is a circuit that reads out the display position data from the sprite attribution table 23. The display position reading out circuit 35 outputs the data which are read out to a frame buffer address generating circuit 36. The frame buffer address generating circuit 36 generates frame buffer addresses based upon the display position data and outputs them to a rendering processing circuit 37. The rendering processing circuit 37 makes rendering processes such as, for example, a rotation of a sprite by 90 degrees. That is, the rendering processing circuit 37 outputs addresses to the sprite buffer 33 to read out the RGB data stored in the sprite buffer 33 in order, and supplies addresses to the frame buffer 39 to write the RGB data read out from the sprite buffer 39 into the frame buffer 39. On this occasion, if the rendering process is not designated, the address outputted from the frame buffer address generating circuit 36 is outputted to the frame buffer 39 as it is. Meanwhile, if the rendering process is designated, the address outputted from the frame buffer address generating circuit 36 is converted in accordance with a predetermined rendering rule and is outputted to the frame buffer 39. Thereby, processes, for example, the rotation process of the sprite, are executed.

[0018]

As thus discussed, pattern data, which belong to the most subordinate sprite in priority read out first from the sprite attribution table, are decompressed and converted to the RGB data,

then the rendering process is made on them and they are written into the frame buffer 39. Next, attribution data of the second sprite are read out from the sprite attribution table 23, and sprite pattern data of the second sprite are read out from the pattern ROM 25 based upon the sprite attribution data. The data being read out are decompressed and converted to the RGB data, then the rendering process is made on them and they are written into the frame buffer 39. On this occasion, the data are overwritten to the first sprite which has been written before. Afterwards, the third sprite, the fourth sprite and further following sprites are written into the frame buffer. When all of the sprites that are predetermined are written into the frame buffer 39, the RGB data stored in the frame buffer 40 are read out in order and in accordance with the addresses generated by a display scan timing control circuit 40 and also are outputted to the display device 26 based upon display scan timing. Thereby, the sprites stored in the pattern ROM 25 are displayed.

[0019]

In the structure discussed above, the sprite buffer 33 and the frame buffer 39 each has a two-buffer structure, i.e., when data are written into one of the buffers, data stored in the rest buffer are read out.

[0020]

As thus described, according to the embodiment, because the compressed pattern data are used, the transferred data amounts per sprite from the pattern ROM 25 decrease and the number of sprite pictures increases. Because the real-time decompression processing circuit 30 for the compressed data and the sprite buffer 33 for storing the decompressed data are built in, the speed at which the sprite pattern data can be transferred to the frame buffer 39 can be easily made faster.

[0021]

For example, in such a case that the pattern data are compressed to 50% thereof, if the transferring speed between the frame buffer 39 and the sprite buffer 33 is doubled, the reading out speed of the pattern data can be doubled in total. Accordingly, unless the data amounts of the sprites are changed, the number of display of the sprites can be doubled. If the sprite buffer 33 and the frame buffer 39 are configured with SRAM or DRAM which can operate rapidly, fast random access can be made. Therefore, the particular process (rendering process) that forms pictures while the sprites being altered such as, for example, being rotated, can be rapidly made.

[0022]

Because the sprite buffer 33 and the frame buffer 39 are built in, the same effect as that of expansion of the bus width made relatively freely can be obtained without requiring cost increase. Thus, the expanded data can be transferred to the frame buffer 39 during the period of real-time expanding time of a specific sprite. Thereby, a bottleneck in the process time for tracing the route: the pattern ROM 25 → the real-time decompression processing circuit 30 → the sprite buffer 33 → the frame buffer 39, can be dissolved. The number of the sprite displays can thus increase in accordance with a compression ratio.

[0023]

As an image compression system that is suitable for the real-time expansion, a compression manner using the run-length compression and line reference. In the run-length compression and the line reference, a line buffer for one line is used; thereby, basically the expansion process can be reduced to a copy process of the read out data for simplification.

[0024]

[Effect of the Invention]

As thus described, according to the present invention, the compressed sprite pattern data are stored into the memory and the

stored data are read out to be decompressed and to be displayed. Therefore, the effect of improving the sprite display capacity without requiring large cost increase or without damaging the reliability can be obtained. Also, according to the invention recited in Claim 3, the writing means alters the storing area of the frame buffer, which is designated by the sprite attribution data, in accordance with the predetermined rule to write the data outputted from the sprite buffer into the altered area. Thus, the further effect of making the particular process such as, for example, the rotation of the sprite to be displayed, can be obtained.

[Brief Description of the Drawings]

FIG. 1 is a block diagram showing a structure of one embodiment of the invention.

FIG. 2 is a block diagram showing a structure of a display unit using an image display device according to the embodiment.

FIG. 3 is a block diagram showing a structure of a conventional image display device.

[Description of Reference Numerals]

- 21: image display device
- 22: CPU
- 23: sprite attribution table
- 25: pattern ROM
- 26: display device
- 30: real-time decompression processing circuit
- 33: sprite buffer
- 37: rendering processing circuit
- 39: frame buffer

FIG. 1

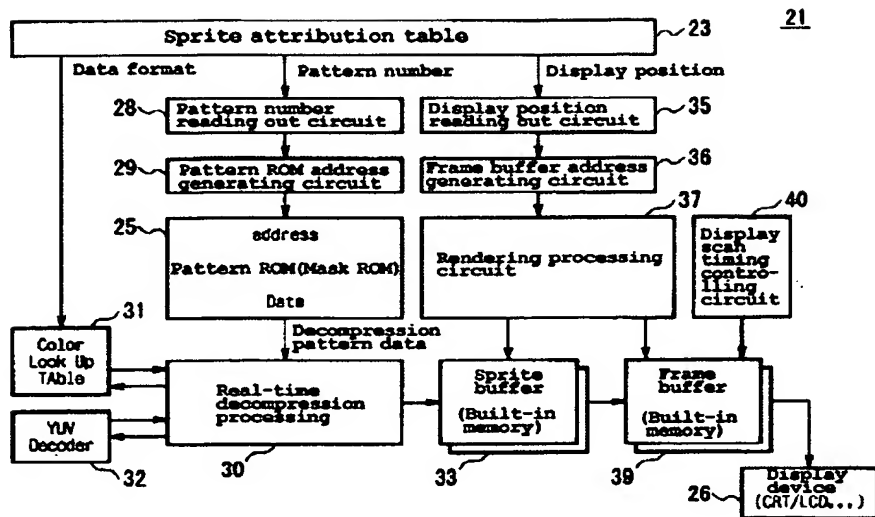


FIG. 2

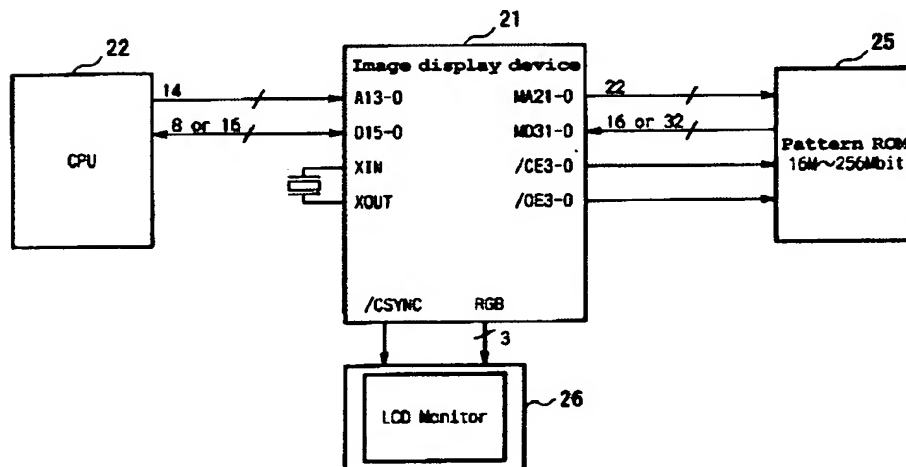
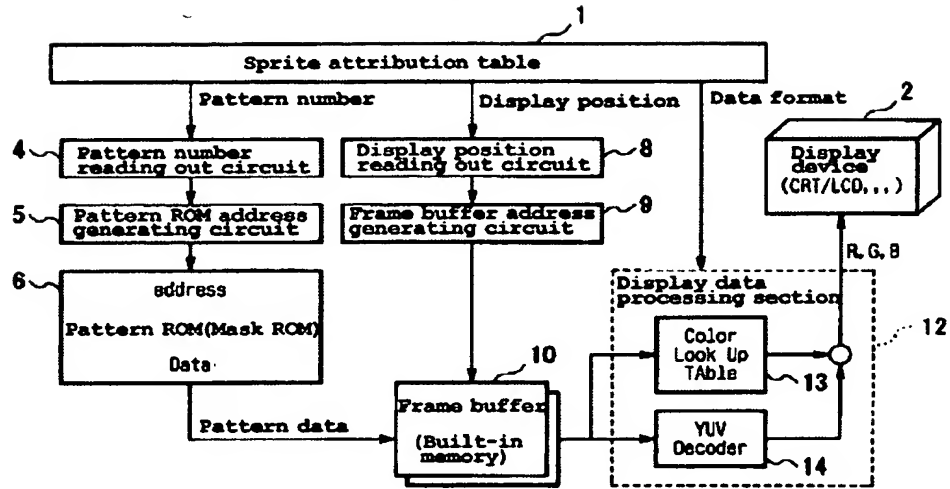


FIG. 3







## 【特許請求の範囲】

【請求項1】 スプライト属性データが書き込まれるスプライト属性テーブルと、

メモリ内の圧縮されたスプライトパターンデータを、前記スプライト属性テーブルが指示する順に読み出す読出手段と、

前記メモリから読み出されたスプライトパターンデータを圧縮前のデータに解凍する解凍手段と、

前記解凍手段によって解凍されたデータが書き込まれるスプライトバッファと、

前記スプライトバッファ内のデータを読み出し、前記スプライト属性データによって指示されるフレームバッファの記憶位置に書き込む書込手段と、

前記フレームバッファ内のデータを表示する表示手段と、

を具備することを特徴とする画像表示装置。

【請求項2】 前記解凍手段は、前記スプライト属性テーブルから読み出されるデータフォーマットに従って前記解凍されたデータをさらにRGBデータに変換して出力することを特徴とする請求項1に記載の画像表示装置。

【請求項3】 前記書込手段は、前記スプライト属性データによって指示されたフレームバッファの記憶位置を所定の規則に従って変更して、前記スプライトバッファの出力データを書き込むことを特徴とする請求項1または請求項2に記載の画像表示装置。

【請求項4】 前記書込手段は、上書きによって前記フレームバッファの書き込みを行うことを特徴とする請求項1～請求項3のいずれかの項に記載の画像表示装置。

## 【発明の詳細な説明】

## 【0001】

【発明の属する技術分野】この発明は、スプライト表示性能の向上を図った画像表示装置に関する。

## 【0002】

【従来の技術】周知のように、ビデオゲーム等の分野においてはスプライト表示方式がしばしば用いられる。ここで、スプライト表示方式とは、画面に表示されるキャラクタ毎に表示位置属性（スプライト属性）を持ち、そのスプライト属性に従いキャラクタを配置することで全体の画面を構成する方式である。表示位置などの属性を持ったキャラクタをスプライトという。ビデオゲームなどのように、インタラクティブに高速にキャラクタを動かす場合、スプライト表示方式では動かすキャラクタのスプライト属性を変更するだけで画面の書き換えをすることができる。

【0003】図3は従来のスプライト表示方式による画像表示装置の構成を示すブロック図である。スプライト属性テーブル1には、表示装置2に表示するスプライト毎の表示位置、スプライトパターン（キャラクタの絵柄）番号、スプライトパターンデータ（絵柄データ）の

データフォーマット等のスプライトに関する属性データが格納されている。このスプライト属性テーブル1に格納されたスプライトパターン番号を基に、パターン番号読み出し回路4およびパターンROMアドレス生成回路5がパターンデータが格納されているパターンROM6のアドレスを生成する。

【0004】一方、表示位置読み出し回路8は、スプライト属性テーブル1に格納されているスプライト表示位置データを読み出し、フレームバッファアドレス生成回路9へ出力する。フレームバッファアドレス生成回路9は、そのスプライト表示位置データに基づいてフレームバッファアドレスを生成し、フレームバッファ10へ出力する。このアドレスによって、パターンROM6から読み出されたパターンデータが、フレームバッファ10に格納される。

【0005】表示されるスプライトの数だけ、上述したパターンデータの読み出し、フレームバッファ10への書き込み処理を行い、1つの画面データがフレームバッファ10に展開される。フレームバッファ10に格納されたパターンデータは、表示装置2のスキヤンタイミングに従い、フレームバッファ10から読み出され、表示データ処理部12に出力される。表示データ処理部12は、スプライト属性テーブルから読み出されたパターンデータフォーマットに従ってカラーlookupテーブル13またはYUVデコーダ14へパターンデータを出し、これによりパターンデータをRGB（レッド・グリーン・ブルー）データに変換し、表示装置2へ出力する。

## 【0006】

【発明が解決しようとする課題】ところで、上述した画像表示装置において、スプライトの表示数はパターンROM6のアクセススピードによって制限される。すなわち、フレームバッファ10は表示装置2のスキヤン周期に合わせてリフレッシュ（画像データの書き換え）が行われる。通常1フレームは60Hzであり、マスクROMのアクセススピードを125ns（8MHz）でバス幅を8bitとすると、1フレームで転送できるデータ量は、

$8\text{MByte}/60\text{Hz} = 133\text{Kbyte}$ となる。スプライトの色数を16色（1ドットあたり4ビットのデータ量）とすると、 $133\text{K} \times 8 / 4 = 266 \times 1024$ ドット（約272000ドット）

となり、NTSCにおける一般的解像度320×240とすると、

$272000 / (320 \times 240) = \text{約} 3.5$ 面分のスプライト表示能力となる。これは、バックグラウンドの絵柄（これもスプライトで構成）にキャラクタを動かすようなアニメーション表示を想定した場合、スプライトが互いに重なり合うことを考慮に入れると十分な表示

能力とは言えない。

【0007】そこで、表示能力をあげる工夫が必要となる。表示能力を上げるためには以下のような方法がある。

(a) パターンROM6とのインタフェースのバス幅を拡げる。

(b) 高速転送が可能な、例えばSDRAMなどにパターンROM6のパターンデータをダウンロードしてアクセスサイクルを高速化する。

【0008】しかしながら、上述した(a)の方法では、別チップのパターンROM6とのインタフェースの端子数が増えコストアップとなる問題があった。また、パターンROM6も並列に複数個のROMを使用する必要があり、これもコストアップにつながった。また、

(b)の方法では、SDRAMを外付けする必要がありSDRAM分のコストアップが生じる。またSDRAMとのインターフェースが必要となりその分の端子数が増え、さらに、パターンデータを揮発性メモリ(SDRAM)に転送することにより、システムの信頼性が損なわれる。すなわち、静電気などによる外乱ノイズによって誤動作を生じる可能性があり、また、揮発性メモリではデータ化けが起こるなどの問題が生じる。SDRAMを内蔵する方法も考えられるが、シナリオ全体のパターンは64Mbitから256Mbit必要で現実的ではない。また、シーン毎に使用するパターンのみをダウンロードして内蔵するSDRAMの容量を少なくする方法もあるが、シーン毎にダウンロードが必要であることや、1シーンで利用できるパターン数が限定されるなどの問題がある。

【0009】この発明は、このような事情を考慮してなされたもので、その目的は、大幅なコストアップを生じることがなく、かつ、信頼性を損なうこともなくスプライト表示能力を上げることができる画像表示装置を提供することにある。

【0010】

【課題を解決するための手段】この発明は上記の課題を解決すべくなされたもので、請求項1に記載の発明は、スプライト属性データが書き込まれるスプライト属性テーブルと、メモリ内の圧縮されたスプライトパターンデータを、前記スプライト属性テーブルが指示する順に読み出す読出手段と、前記メモリから読み出されたスプライトパターンデータを圧縮前のデータに解凍する解凍手段と、前記解凍手段によって解凍されたデータが書き込まれるスプライトバッファと、前記スプライトバッファ内のデータを読み出し、前記スプライト属性データによって指示されるフレームバッファの記憶位置に書き込む書込手段と、前記フレームバッファ内のデータを表示する表示手段とを具備することを特徴とする画像表示装置である。

【0011】また、請求項2に記載の発明は、請求項1に記載の画像表示装置において、前記解凍手段は、前記

スプライト属性テーブルから読み出されるデータフォーマットに従って前記解凍されたデータをさらにRGBデータに変換して出力することを特徴とする。また、請求項3に記載の発明は、請求項1または請求項2に記載の画像表示装置において、前記書込手段は、前記スプライト属性データによって指示されたフレームバッファの記憶位置を所定の規則に従って変更して、前記スプライトバッファの出力データを書き込むことを特徴とする。

【0012】また、請求項4に記載の発明は、請求項1～請求項3のいずれかの項に記載の画像表示装置において、前記書込手段は、上書きによって前記フレームバッファの書き込みを行うことを特徴とする。

【0013】

【発明の実施の形態】以下、図面を参照し、この発明の一実施の形態について説明する。図1は同実施の形態による画像表示装置21の構成を示すブロック図、図2は同画像表示装置21およびCPU(中央処理装置)22によって構成された表示ユニットの構成を示すブロック図である。図2において、パターンROM25(マスクROM)には、複数のスプライトのパターンデータが、例えばランレングス圧縮方式のようにリアルタイム解凍が可能な圧縮方式によって圧縮され、記憶されている。CPU22は、パターンROM25内の各スプライトを表示装置26の表示画面のどの位置に表示させるかを示す表示位置データやその他のスプライト属性データを画像表示装置21へ出力する。画像表示装置21はパターンROM25から各スプライトのパターンデータを読み出し、リアルタイム解凍を行い、さらにRGBデータに変換し、CPU22からの位置データに従って表示装置26に表示させる。

【0014】以下詳述すると、図1において、符号23はRAM(ランダムアクセスメモリ)によって構成されたスプライト属性テーブルであり、パターンROM25に記憶されているスプライトの属性を示す属性データが、スプライト毎にCPU22によって書き込まれる。この属性データには、スプライトパターンの番号を示すパターン番号、スプライトの表示画面上の表示位置を指定する表示位置データ、スプライトパターンデータのデータのフォーマット、使用するカラーパレットの番号を指示するパレット番号等がある。

【0015】28は、スプライト属性テーブル23内のパターン番号を、表示すべき優先順位の下のものから順次読み出すパターン番号読み出し回路である。29はパターン番号読み出し回路28から読み出されたパターン番号を受けて、パターンROM25からスプライトを読み出すためのアドレスを生成するパターンROMアドレス生成回路であり、生成したアドレスをパターンROM25へ出力する。パターンROM25へ上記アドレスが供給されると、パターンROM25からスプライトのパターンデータが読み出され、リアルタイム解凍処理回

路 30 へ出力される。

【0016】リアルタイム解凍処理回路 30 は、パターン ROM 25 から読み出された圧縮パターンデータを圧縮前のパターンデータにリアルタイムで解凍し、解凍済みのデータをカラーlookupアップテーブル 31 または YUV デコーダ 32 によって RGB データに変換し、スプライトバッファ 33 に書き込む。ここで、カラーlookupアップテーブル 31 または YUV デコーダ 32 のいずれが使用されるかは、スプライト属性テーブル 23 から読み出されるデータフォーマットによって決められる。すなわち、圧縮前のパターンデータのフォーマットがカラーコードであった場合はカラーlookupアップテーブル 31 が使用され、YUV データであった場合は YUV デコーダ 32 が使用される。

【0017】一方、表示位置読み出し回路 35 は、スプライト属性テーブル 23 から表示位置データを読み出す回路であり、読み出したデータをフレームバッファアドレス生成回路 36 へ出力する。フレームバッファアドレス生成回路 36 は、表示位置データに基づいてフレームバッファアドレスを生成し、レンダリング処理回路 37 へ出力する。レンダリング処理回路 37 はスプライトを 90 度回転させる等のレンダリング処理を行う。すなわち、このレンダリング処理回路 37 は、スプライトバッファ 33 へアドレスを出力してスプライトバッファ 33 内の RGB データを順次読み出すと共に、フレームバッファ 39 へアドレスを供給して、スプライトバッファ 33 から読み出された RGB データをフレームバッファ 39 に書き込む。この際、もし、レンダリング処理が指定されていなかった場合は、フレームバッファアドレス生成回路 36 から出力されたアドレスをそのままフレームバッファ 39 へ出力し、また、レンダリング処理が指定されていた場合は、フレームバッファアドレス生成回路 36 から出力されたアドレスを所定のレンダリング規則に従って変換し、フレームバッファ 39 へ出力する。これにより、例えば、スプライトの回転処理等が行われる。

【0018】このようにして、スプライト属性テーブルから最初に読み出された最も優先順位の低いスプライトのパターンデータが解凍され、RGB データに変換され、さらにレンダリング処理され、フレームバッファ 39 に書き込まれる。次に、2 番目のスプライトの属性データがスプライト属性テーブル 23 から読み出され、このスプライト属性データに基づいて 2 番目のスプライトパターンデータがパターン ROM 25 から読み出される。そして、読み出されたデータが解凍され、RGB データに変換され、レンダリング処理され、フレームバッファ 39 に書き込まれる。この時、既に書き込まれている 1 番目のスプライトに上書きされる。以後、3 番目、4 番目・・・のスプライトがフレームバッファに書き込まれる。そして、予め決められた全スプライトがフレー

ムバッファ 39 に書き込まれると、表示スキャンタイミング制御回路 40 が発生するアドレスにしたがってフレームバッファ 40 内の RGB データが順次読み出され、表示スキャンタイミングに基づいて表示装置 26 へ出力される。これにより、パターン ROM 25 内のスプライトが表示される。

【0019】上記の構成において、スプライトバッファ 33、フレームバッファ 39 は各々 2 バッファ構成であり、一方のバッファに書き込まれている時、他方のバッファ内のデータが読み出される。

【0020】以上説明したように、上記実施形態によれば、パターンデータに圧縮データを用いることにより、1 スプライト当たりのパターン ROM 25 からのデータ転送量が減りスプライト描画数が増える。圧縮データのリアルタイム解凍処理回路 30 と、解凍後のデータを保持するスプライトバッファ 33 を内蔵することで、フレームバッファ 39 へ転送できるスプライトパターンデータの転送速度を上げることが容易になる。

【0021】例えば、パターンデータを 50 % に圧縮した場合、フレームバッファ 39 とスプライトバッファ 33 間の転送スピードを 2 倍に増やせば、トータルでパターンデータの読み出し速度を 2 倍に増やすことができ、したがって、スプライトのデータ量が同一の場合、スプライトの表示数を 2 倍に増やすことができる。スプライトバッファ 33、フレームバッファ 39 は高速の SDRAM や SDRAM などにより構成することにより高速にランダムアクセスが可能となり、スプライトを回転などの変形処理を行いながら画面を構成する処理（レンダリング処理）を高速に行うことが可能となる。

【0022】また、スプライトバッファ 33 とフレームバッファ 39 を内蔵することにより、コストアップなしにバス幅を比較的自由に広げたと同一の効果を得ることができ、スプライトのリアルタイム伸張時間内に伸張済みのデータをフレームバッファ 39 へ転送することが可能となる。これにより、パターン ROM 25 →リアルタイム解凍処理回路 30 →スプライトバッファ 33 →フレームバッファ 39 の経路での処理時間のネックが解消され圧縮率に従ったスプライト表示数の増大が可能となる。

【0023】また、リアルタイム伸張に適した画像圧縮方式として、ランレングス圧縮とライン参照を用いた圧縮方法を用いてもよい。ランレングス圧縮およびライン参照では 1 ライン分のラインバッファを使用することによって基本的に読み出しデータのコピーのみに伸張処理を単純化することが可能となる。

【0024】

【発明の効果】以上説明したように、この発明によれば、メモリ内に圧縮されたスプライトパターンデータを記憶させ、該データを読み出して解凍し、表示するようにしたので、大幅なコストアップを生じることがなく、

かつ、信頼性を損なうことなくスプライト表示能力を上げることができる効果が得られる。また、請求項3に記載の発明によれば、書込手段は、スプライト属性データによって指示されたフレームバッファの記憶位置を所定の規則に従って変更して、スプライトバッファの出力データを書き込むので、例えばスプライトを回転させて表示する等スプライトに特定の処理を施すことができる効果がある。

【図面の簡単な説明】

【図1】 この発明の一実施形態の構成を示すブロック図である。

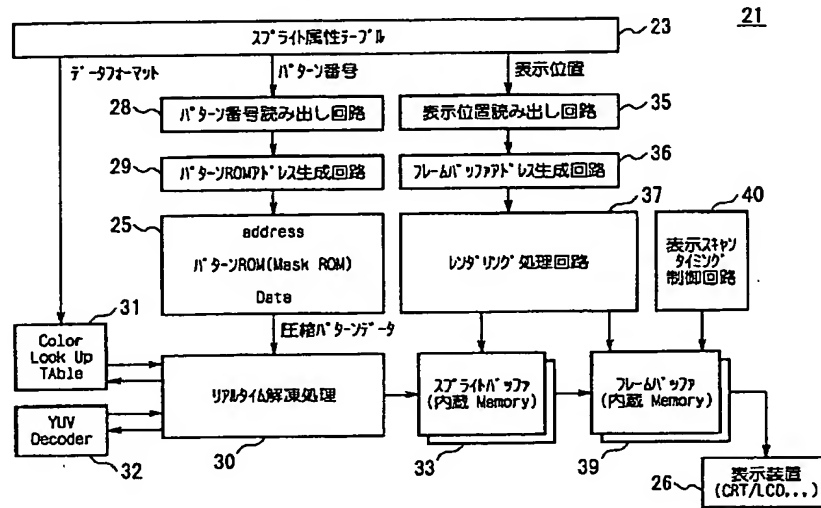
【図2】 同実施形態による画像表示装置を用いた表示ユニットの構成を示すブロック図である。

【図3】 従来の画像表示装置の構成を示すブロック図である。

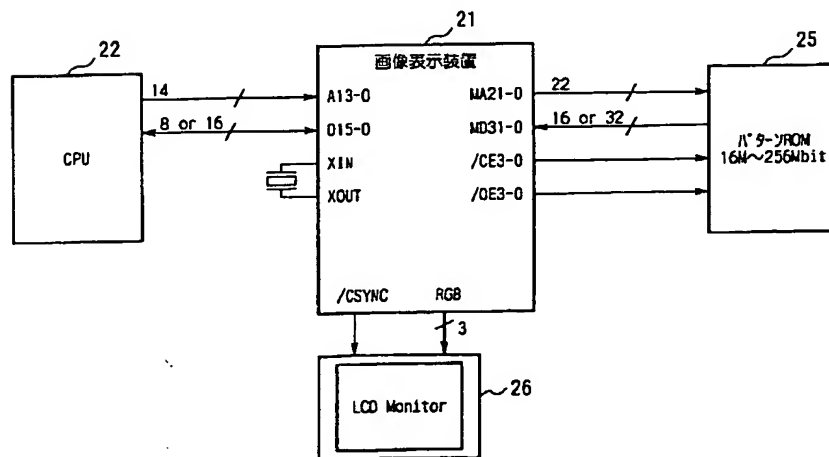
【符号の説明】

21…画像表示装置、22…CPU、23…スプライト属性テーブル、25…パターンROM、26…表示装置、30…リアルタイム解凍処理回路、33…スプライトバッファ、37…レンダリング処理回路、39…フレームバッファ。

【図1】



【図2】



【図3】

